Reliability Analysis of Fault-Tolerant Reconfigurable Nano-Architectures

(Categories: High Level Design Error Modeling, Fault Tolerance)

Debayan Bhaduri
ISR-3
Los Alamos National Laboratory
Los Alamos, NM
Email: dbhaduri@lanl.gov

Paul Graham
ISR-3
Los Alamos National Laboratory
Los Alamos, NM
Email: grahamp@lanl.gov

Sandeep K. Shukla
Fermat Lab
Virginia Tech
Blacksburg, VA 24061
Email: shukla@vt.edu

***Abstract*— Manufacturing and transient faults may be abundant in high density reconfigurable design fabrics built with nano-scale technologies (silicon or other emerging technologies). Design of reliable digital logic and architectures on such defective fabrics will require adequate redundancy. However, analysis of redundancy/reliability trade-offs for such designs will be required for micro-architects to do design space explorations. An automated computational scheme based on Markov Random Fields (MRFs) and Belief Propagation techniques was incorporated in a tool named NANOLAB to compute these trade-offs in the face of thermal perturbations and interconnect noise. However, previously this tool was used only for combinational design exploration. In this paper, we show how this tool and the methodology can be extended to analyze defect-tolerant programmable sequential logic design. The effectiveness of this automation is illustrated by analyzing reconfigurable Boolean networks formed by using different industry standard configurable logic blocks (CLBs) in the presence of thermal and signal noise.** [1]

I. Introduction

With the advent of nanotechnology, it is desirable that digital systems exhibit dynamic defect-tolerant attributes. It has been shown in [10] and [13] that reconfigurable logic architectures like field-programmable gate arrays (FPGAs) may mitigate both manufacturing and transient defects common to nano-substrates. Also, [11] analyzes the NAND multiplexing [14] and reconfiguration fault-tolerant techniques, and presents a defect- and fault-tolerant architecture in which multiplexing (with a low degree of redundancy) is combined with a massively reconfigurable architecture. This points out the fact that different degrees of redundancy need to be applied at different granularity levels (such as gate level, configurable logic block (CLB) level, etc. [6]) to make systems cost-effectively reliable. Our goal is to map Boolean functions onto reconfigurable logic blocks with adequate redundancy so that the resulting logic network computes the intended function with higher reliability. In order to achieve this, in-depth analysis is required to find suitable redundancy and granularity levels for specific reliability measures of such architectural configurations.

Main Results: In previous work [2], [5], the MRF based model of computation (discussed in detail later) has only been used to analyze combinational Boolean networks. In this work, we have implemented a loopy Belief Propagation algorithm [8] so that sequential circuits (such as sequential CLBs) can also be analyzed with our tool NANOLAB. We have incorporated capabilities to configure logic networks dynamically and to dynamically introduce faults such as stuck-at faults or single-event upsets (SEUs). These NANOLAB enhancements are major contributions enabling dynamic fault injection and the modeling of reconfigurable logic. It has been stated by experts that due to their regularity and simplicity, reconfigurable logic architectures may be some of the earliest programmable architectures implemented using nano-scale technology. Thus, in this paper, we show how different configurable blocks can be modeled with NANOLAB. We are also in the process of modeling complex networks using CLBs that are widely used (Xilinx, Actel). We claim that such an attempt to analyze granularity and redundancy levels of reconfigurable nano-architectures has not been undertaken in the past. [13] and [10] propose methodologies to locate manufacturing defects in the reconfigurable fabrics and form accurate defect maps which encompass all the fault locations. Such maps can be used to layout circuits on the fabrics so as to avoid the defective devices. We suggest a methodology wherein a Boolean network is mapped onto a defect-prone reconfigurable nano-architecture with suitable redundancy either at the application, CLB or gate levels, such that higher reliability of computation is achieved in the presence of faults (both static and dynamic).

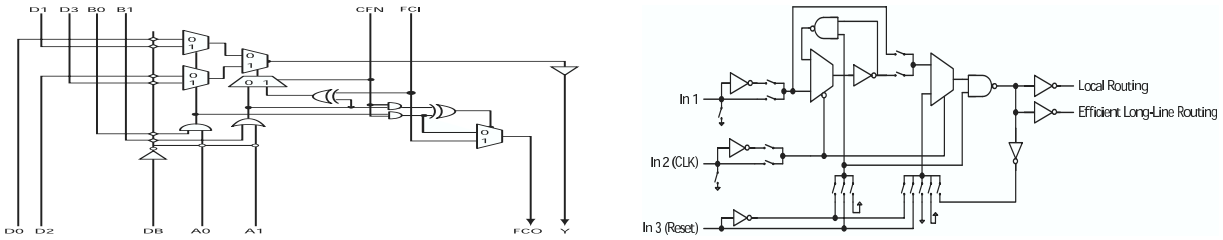
II. Background

MRF-Based Methodology: The basis for the approach in [2] is based on Markov Random Fields. An MRF is defined as a finite set of random variables, $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_k\}$. Each

[1]

Contact Author Information
Name: Debayan Bhaduri
Address: ISR-3, Space Data Systems
Los Alamos National Laboratory, Los Alamos, NM
Email: dbhaduri@lanl.gov **Tel:** (505) 665-0019 **Fax:** (505) 665-4197

LA-UR-04-4446



(a) C-cell from Axcelerator family FPGAs

(b) Core Logic Tile from ProAsic Flash family FPGAs

Fig. 1. Configurable Logic Blocks (CLBs) from Actel [1]

variable λ_i has a neighborhood, N_i, which has variables from $\{\Lambda$ - $\lambda_i\}$. The probability distribution of a given variable depends only on a (typically small) neighborhood of other variables that is called a clique. Due to the Hammersley-Clifford theorem [4],

$$\mathbf{P}(\lambda_i|\{\mathbf{\Lambda}-\lambda_i\}) = \frac{\mathbf{1}}{\mathbf{Z}}\mathbf{e}^{\frac{-1}{KT}\sum_{c\in C}U_c(\lambda)} \quad (1)$$

The conditional probability in equation 1 is the Gibbs distribution. Z is the normalizing constant and for a given node i, C is the set of cliques. U_c is the clique energy function [2] and depends only on the neighborhood of the node whose energy state probability is being calculated. The logic margins of nodes in a Boolean network decrease at higher values of *KT* and increase at lower values. The logic margin in this case is the difference between the probabilities of occurrence of a logic low and a logic high. Higher logic margins result in better reliability of computation. This formulation also allows correct analysis of entropy values, since the entropy in the system is inversely proportional to the logic margin. Thus, this methodology not only provides a different non-discrete model of computation, in fact, it relates information theoretic entropy and thermal entropy of computation in a way so as to connect reliability to entropy (discussed in detail later). It has been shown that the thermodynamic limit of computation is $KT\ln 2$ [3] where *KT* is the thermal energy (*K* is the Boltzmann constant and T is the temperature in Kelvin) and is expressed in normalized units relative to the logic energy (clique energy). The thermodynamic limit of computation is the thermal energy that is proportional to the minimum entropy loss due to irreversible computation. If we consider energy levels close to these thermal limits, the reliability of computation is likely to be affected. The model of computation in [2] considers thermal perturbations, discrete errors and continuous signal noise [7] as sources of errors. The idea is to use a Gibbs distribution based technique to characterize the logic computations by Boolean gates and represent logic networks as MRFs and maximize probability of being in valid energy configurations at the outputs.

Defect- and Fault-Tolerance through Reconfiguration: A computer architecture that can be configured or programmed after fabrication to implement desired computations is said to be *reconfigurable*. Reconfigurable fabrics such as FPGAs are composed of programmable logic elements (often referred to as CLBs) and interconnects, and these can be programmed or configured to implement any circuit. Defect-tolerance can be achieved in FPGA-like architectures by detecting faulty components during testing and excluding them during reconfiguration. It is expected [13] that reconfigurable fabrics made from next generation manufacturing techniques will go through a post-fabrication testing phase during which these fabrics will be configured for self-diagnosis. The test circuits placed on the fabric during this self-diagnosis phase will utilize resources that will be available later for normal fabric operation. The testing can be done with massive parallelism, drastically reducing test time. While such reconfigurable architectures may aid in circumventing manufacturing defects at the nano-scale, architectures such as the Cell Matrix [10] have been proposed to support dynamic defect tolerance.

In this study, we are analyzing some simple configurable logic block structures for their defect- and fault-tolerance in nano-scale implementations. Figure 1 shows the two specific CLBs analyzed for this study that are found in commercial Actel FPGAs. Note that these CLBs are examples of reconfigurable core logic, and, eventually, we plan to analyze more of these. The C-cell in Figure 1(a) is from the Axcelerator (AX) anti-fuse FPGA family while the core tile shown in Figure 1(b) is from the ProAsic^{PLUS} flash FPGA family (see [1] for more information on these FPGAs). Our analysis of these architectures are not concerned with a specific memory technology for holding the programming data but, rather, how the basic logic cells can be made more reliable for nano-scale implementation using redundancy at various architectural levels.

Loopy Belief Propagation: The computation of posterior marginals on nodes in an arbitrary Bayesian or Markov Random network is a NP-hard problem [9]. Different Belief Propagation algorithms and approximation schemes have

been proposed in the past, and attempts have been made to categorize different types of networks for which each algorithm works best. In this work, we have implemented a technique called *Unwrapped tree* [8] to analyze loops in the sequential portions of the ProAsic^{PLUS} core logic tile (shown in Figure 1(b)). To illustrate the technique, let us walk through an example. Figure 2 shows a network G that forms an undirected cycle composed of four nodes {1, 2, 3, 4}. To analyze the marginal probability values at each of these nodes, G is translated to T which is the corresponding unwrapped tree. T is an acyclic graph that is locally equivalent to the original graph, G.

The unwrapping technique is as follows: choose an arbitrary node r and initialize T = r (in Figure 2, node 1 is chosen). For each leaf node m of T, find the neighbors of the corresponding node in G, other than the parent of m in T. Add these nodes to the tree. The probability distributions at the nodes of both G and T remain same. The probability values at the leaves of T are compared. When they become equivalent (approximation thresholds may be defined), the beliefs are said to have converged. The cycle is unwrapped till convergence is achieved, but it has also been observed in [9] and this work that when loops are present in networks, messages may circulate indefinitely around the loops and the algorithm may not converge to a steady state. Thus, a restriction may be imposed on the depth of the tree T (number of times the loop in the Boolean network is unwrapped) such that the algorithm does not form an infinite chain.

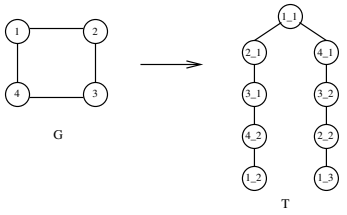


Fig. 2. Unwrapped Tree - Technique for Analyzing Loopy Belief Propagation

III. NANOLAB and Our Methodology

NANOLAB [5] is a MATLAB based reliability evaluation tool, that uses entropy as the reliability metric. In the context of this paper, we define entropy as the measure of the disorder of a system. It is considered to have high values when the system under consideration is very disordered (hence unreliable). Let us consider a random variable X, which must take on one of the values $x_1, x_2, ..., x_n$ with respective probabilities $p_1, p_2, ..., p_n$. Then, the expected degree of uncertainty (randomness) in the system that is dependent upon X is: $H(X) = -\sum_i p_i \cdot log(p_i)$. This is information entropy of the random variable X, which can be interpreted as the average amount of uncertainty associated with the random variable X.

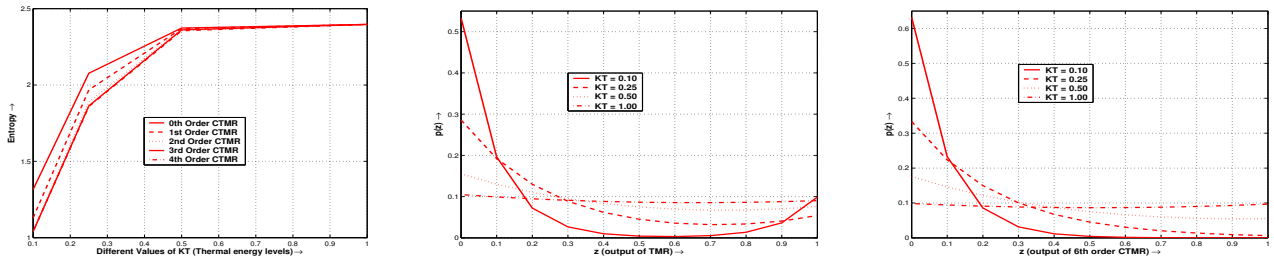
NANOLAB automates the MRF-based methodology briefly discussed in Section II. It consists of a library of functions and a Belief Propagation algorithm [12] that can compute energy distribution and entropy at the primary/intermediate outputs and interconnects of arbitrary Boolean networks, given discrete or continuous (signal noise) energy distributions at the primary inputs and interconnects of the circuits. These functions work for any generic one-, two- and three- input logic gates and can be extended to handle n-input logic gates as well as take in as inputs the logic compatibility function (similar to truth table) [2] and the initial energy distribution for the inputs of a gate. Energy distributions are returned as vectors by these functions and indicate the probability of the output of a gate being at different energy levels between 0 and 1.These probabilities are also calculated over different values of *KT* to analyze thermal effects on the node. NANOLAB also consists of functions that can model noise either as uniform or Gaussian distributions or combinations of these, depending on the user specifications. Arbitrary Boolean networks in any redundancy-based fault-tolerant architectural configuration can be analyzed by writing simple MATLAB scripts that use these NANOLAB library functions.

We have enhanced the capabilities of our tool by developing libraries for the core CLBs in Figure 1. The dynamic programming of these core logic blocks are controlled by configuration files that are given as inputs to these libraries along with energy distributions at the inputs of these CLBs. Errors can also be introduced in these configuration files either interactively or by using some specific error distribution. Entropy values and energy distributions at the outputs of the CLBs are returned by the libraries. For modeling the sequential part of the core logic tile shown in Figure 1(b), we have implemented the *Unwrapped tree* loopy Belief Propagation algorithm discussed in Section II. But due to external posterior marginal probabilities (dependencies) in the loop of the logic tile, the Belief Propagation algorithm sometimes does not converge within the threshold number of iterations. We are looking at approximation schemes to solve this problem. We have also used loop unrolling to implement the sequential portion of this logic block, and this technique seems to work better but with lesser degree of probabilistic accuracy.

With such a framework, we expect to analyze different reliability-redundancy trade-off points at different levels of granularity [6] such as at the gate, CLB level or the application levels. For instance, at the application level, multiple CLBs may be used to implement a redundant function with voting; at the CLB level, the architecture itself may triplicate CLBs and vote on their outputs; and, at the gate level, individual gates or components within the CLB may be made redundant. As our tool has been augmented with the capability to handle sequential circuits, we can analyze complex systems with NANOLAB. Both thermal perturbations and signal noise can be introduced in the FPGA models, and different hardware redundancy based techniques may be adopted within the reconfigurable architectural framework.

IV. Experiments and Results

Reliability and Entropy Measures of Axcelerator CLB: Figure 3 (a) indicates the entropy values when the C-cell



(a) Entropy at different KT values for Actel AX C-cell

(b) Output distribution of a TMR for Actel ProAsic CLB

(c) Output distribution of 6th order CTMR for Actel ProAsic CLB

Fig. 3. Entropy and Energy distribution at the outputs of different CLBs

is configured to perform a two-input OR function. The entropy values are plotted through the 4th order CTMR for different KT values. It can be observed that as redundancy is increased by adding more CTMR orders, the entropy decreases (logic margin and reliability increases) at lower KT values. However, the rate of improvement in reliability decreases once the 2nd order CTMR is reached, and further augmentation of redundant devices does not improve the reliability of computation appreciably. It can also be seen that the 3rd and 4th order CTMR have almost equivalent entropy values at different thermal energy levels. This result can be interpreted as follows: for a certain redundancy level, the system's reliability for a given configuration reaches a steady state. Any further increase in redundancy may either marginally improve the reliability or even worsen it.

Energy distributions at the Output of the ProAsic CLB: NANOLAB can also be used to compute the probability of different energy configurations and, thus, the reliability at the primary outputs of a Boolean network. Figure 3 (b) and (c) show the energy distributions at the outputs of a TMR and a 6th order CTMR configuration applied to the ProAsic core logic tile functioning as an OR gate, respectively. Note that the probability values are based on bin sizes of 0.1. It can be seen that the logic margins for the output (z) at KT values of 0.1, 0.25 and 0.5 are higher for the higher CTMR orders. Also, the probability of z ($p(z)$) being at logic low is higher than being at one because of the configuration and the input distribution for the CLB. It is also observed that at a KT value of one, the logic margin for any CTMR configuration becomes really small (output energy distribution becomes almost uniform) and remains the same even with an increase of redundancy resulting in unreliable computation. Comparing these different orders of CTMR in Figure 3, we infer that for lower thermal energy levels, the probability of being in a valid energy configuration increases as more redundancy is added to the architecture. But further experimental results show that this increase in probability slows down as higher orders of CTMR are reached. This can be understood as follows: the logic margin of the system reaches a saturation point after which reliability can no longer be improved.

In summary, we have shown how we enhanced NANOLAB by augmenting capabilities to analyze reliability-redundancy trade-offs of reconfigurable FPGA-like architectures. Also, we have implemented a loopy Belief propagation algorithm such that NANOLAB can now be used to model sequential circuits and CLBs. This makes our tool more effective in analyzing reliability measures of different sequential reconfigurable Boolean networks.

REFERENCES

[1] *Actel datasheets*. Web Page: `http://www.actel.com/techdocs/ds/default.asp`.

[2] R. I. Bahar, J. Mundy, and J. Chen, *A probability-based design methodology for nanoscale computation*, in *International Conference on Computer-Aided Design* (IEEE Press, San Jose, CA, November 2003).

[3] C. H. Bennett, 'The thermodynamics of computation–a review', *International Journal of Theoretical Physics* **21** (1982.), no. 905-940.

[4] J. Besag, 'Spatial interaction and the statistical analysis of lattice systems', *Journal of the Royal Statistical Society* **Series B** (1994), no. 36(3), 192–236.

[5] D. Bhaduri and S. K. Shukla, *Nanolab: A tool for evaluating reliability of defect-tolerant nano architectures*, in *IEEE Computer Society Annual Symposium on VLSI* (Lafayette, louisiana, February 2004).

[6] D. Bhaduri and S. K. Shukla, *Nanoprism: A tool for evaluating granularity vs. reliability trade-offs in nano architectures*, in *GLSVLSI* (ACM, Boston, MA, April 2004).

[7] D. Bhaduri and S. Shukla, 'Reliability analysis in the presence of signal noise for nano architectures', *IEEE Conference on Nanotechnology* (2004). To Appear.

[8] K. Branson, *Loopy belief propagation*. Available at `http://vision.ucsd.edu/~kbranson/research/bp_pres.pdf`.

[9] G. Cooper, 'The computational complexity of probabilistic inference using bayesian belief networks', in *Artificial Intelligence*, **42** (1990), 393–405.

[10] L. Durbeck and N. Macias, *Defect tolerant, fine-grained parallel testing of a cell matrix*, **4867** (2002).

[11] J. Han and P. Jonker, 'A defect- and fault-tolerant architecture for nanocomputers', *Nanotechnology* (2003), 224230. IOP Publishing Ltd.

[12] J.Yedidia, W.Freeman, and Y.Weiss, *Understanding belief propagation and its generalizations*, in *Proc. International Joint Conference on AI* (2001). Distinguised Lecture.

[13] M. Mishra and S. C. Goldstein, *Defect tolerance at the end of the roadmap*, in *International Test Conference (ITC)* (Charlotte, NC, Sep 30-Oct 2 2003).

[14] J. von Neumann, 'Probabilistic logics and synthesis of reliable organisms from unreliable components', *Automata Studies* (1956), 43–98.